IN THE CLAIMS:

A listing of the status of all claims 1-38 in the present patent application is provided in attached Appendix A.

REMARKS

The Office Action dated April 28, 2004, has been received and carefully considered. Reconsideration of the outstanding rejections in the present application is respectfully requested based on the following remarks.

I. THE ANTICIPATION REJECTION OF CLAIMS 1-38

On page 2 of the Office Action, claims 1-38 were rejected under 35 U.S.C. § 102(e) as being anticipated by Siu (U.S. Patent No. 6,252,851). This rejection is hereby respectfully traversed.

Under 35 U.S.C. § 102, the Patent Office bears the burden of presenting at least a prima facie case of anticipation. In re Sun, 31 USPQ2d 1451, 1453 (Fed. Cir. 1993) (unpublished). Anticipation requires that a prior art reference disclose, either expressly or under the principles of inherency, each and every element of the claimed invention. Id.. "In addition, the prior art reference must be enabling." Akzo N.V. v. U.S.

International Trade Commission, 808 F.2d 1471, 1479, 1 USPQ2d 1241, 1245 (Fed. Cir. 1986), cert. denied, 482 U.S. 909 (1987). That is, the prior art reference must sufficiently describe the claimed invention so as to have placed the public in possession of it. In re Donohue, 766 F.2d 531, 533, 226 USPQ 619, 621 (Fed. Cir. 1985). "Such possession is effected if one of ordinary skill in the art could have combined the publication's description of the invention with his own knowledge to make the claimed invention." Id..

The Examiner asserts that Siu discloses organizing a forward data buffer into one or more queues that store at least one forward data packet. In support of this allegation, the Examiner cites to col. 4, lines 20-36. However, Applicant respectfully submits that the cited portion of Siu does not disclose the claimed feature. The cited portion only makes mention of a queue in the statement that "an estimate is maintained of effective queue size…." No mention is made that this is a forward data buffer or that it stores forward data as claimed by applicant. For at least this reason, Applicant respectfully submits that Siu fails to disclose or suggest the claimed feature of "organizing a forward data buffer

into one or more queues that store at least one forward data packet."

Siu fails to disclose or suggest other claimed features. For example, Applicant respectfully disagrees that Siu discloses or suggests the claimed feature of "calculating the network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues." The Examiner relies on Siu, col. 6, lines 29-67 to allegedly disclose or teach this feature. However, the cited passage states that when window size w is less than some value W_{mid}, the window size is simple increased by one each time an acknowledgement arrives. Therefore, Siu does not disclose a calculation based upon information pertaining to the queue size. For at least these reasons, applicant respectfully submits that the rejection of claim 1 is improper and requests that it be withdrawn.

Claims 11, 18 and 28 recite features similar to the above that are also not disclosed or suggested by Siu. For at least these reasons, applicant respectfully submits that the rejections of claims 11, 18 and 28 are improper and requests that they be withdrawn.

Claims 2-10, 12-17, 19-27 and 29-38 are dependent upon one of independent claims 1, 11, 18 or 28. Thus, since independent claims 1, 11, 18 and 28 should be allowable as discussed above, claims 2-10, 12-17, 19-27 and 29-38 should also be allowable at least by virtue of their dependency on independent claims 1, 11, 18 or 28. Moreover, these claims recite additional features which are not claimed, disclosed, or even suggested by the cited references taken either alone or in combination. For example, some of the dependent claims (e.g., claim 4) recite "calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters." Siu does not disclose or suggest this feature.

In view of the foregoing, it is respectfully requested that the aforementioned anticipation rejection of claims 2-10, 12-17, 19-27 and 29-38 be withdrawn.

II. CONCLUSION

In view of the foregoing, it is respectfully submitted that the present application is in condition for allowance, and an early indication of the same is courteously

solicited. The Examiner is respectfully requested to contact the undersigned by telephone at the below listed telephone number, in order to expedite resolution of any issues and to expedite passage of the present application to issue, if any comments, questions, or suggestions arise in connection with the present application.

To the extent necessary, a petition for an extension of time under 37 CFR § 1.136 is hereby made.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-0206, and please credit any excess fees to the same deposit account.

Respectfully submitted,

Hunton & Williams LLP

By: ______________________
Christopher Cuneo
Registration No. 42,450, for
Thomas E. Anderson
Registration No. 37,063

TEA/CJC

Hunton & Williams LLP
1900 K Street, N.W.
Washington, D.C. 20006-1109
Telephone: (202) 955-1500
Facsimile: (202) 778-2201

Date: January 21, 2005

APPENDIX A

1 (Original). A method for controlling the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, the method comprising:

organizing a forward data buffer into one or more queues that store at least one forward data packet;

calculating the network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

providing the network device's advertised window size to a TCP source; and

calculating a dynamic buffer threshold based, at least in part, upon the sum of the queue sizes and the shared buffer space.

2 (Original). The method of claim 1 wherein the step of organizing a forward data buffer further comprises:

organizing the forward data buffer into one or more queues with one queue per service class.

3 (Original). The method of claim 1 wherein the at least one forward data packet is stored according to its service class.

4 (Original). The method of claim 1 wherein the step of calculating a network device's advertised window size further comprises:

initializing a timer to a predetermined time interval Δt, and an iteration counter to a predetermined initial value n;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δt;

calculating a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters;

resetting the timer, upon expiration of the predetermined interval Δt; and

iterating the iteration counter, upon expiration of the predetermined time interval Δt.

5 (Original). The method of claim 4 wherein the steps of calculating a current error signal $e_i(n)$ and calculating the network device's advertised window size further comprise:

filtering the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha \hat{e}_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

6 (Original). The method of claim 1 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

carrying information relating to the network device's advertised window size by returning TCP acknowledgements in a receiver's advertised window field.

7 (Original). The method of claim 1 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

updating a TCP receiver's advertised window size.

8 (Original). The method of claim 7 wherein the step of updating a TCP receiver's advertised window size further comprises:

identifying whether a packet is an ACK packet, and, if not, putting the non-ACK packet in a reverse data buffer;

determining a service class for the identified ACK packet;

reading the TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

determining whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not setting a advertised window field in the identified ACK packet equal to the network device's advertised window size $W_i(n)$ and

updating the checksum field for the identified ACK packet.

9 (Original). The method of claim 1 wherein the step of calculating a dynamic buffer threshold further comprises:

initializing a timer to a predetermined time interval Δs and an iteration counter to a predetermined initial value n;

setting an initial dynamic buffer threshold $T(0)$ equal to a gain constant γ multiplied by a buffer size B and divided by a number of service classes K;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δs;

calculating a sum of the sampled current queue size according to the equation: $Q(n) = \sum_{i=1}^{K} q_i(n)$;

determining whether the sum of the sampled current queue size is less than the product of the gain constant and the buffer size γB;

if so, updating the dynamic buffer threshold according to $\min\{T(n-1) + \Delta T, \gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes;

if not, updating the dynamic buffer threshold according to $\max\{T(n-1) - \Delta T, T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold;

resetting the timer, upon expiration of the predetermined interval Δs; and

iterating the iteration counter, upon expiration of the predetermined time interval Δs.

10 (Original). The method of claim 9 wherein the step of calculating a sum of the sampled current queue size further comprises:

filtering the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n) = (1-\varphi)\hat{Q}(n-1) + \varphi Q(n)$, wherein φ is a predetermined parameter.

11 (Original). An apparatus for controlling the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, the apparatus comprising:

a forward data buffer, organized into one or more queues that store at least one forward data packet;

a network device's advertised window size calculation module that calculates a network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

a feed back module that provides the network device's advertised window size to a TCP source; and

a dynamic buffer threshold module that calculates a dynamic buffer threshold based, at least in part, upon the sum of the queue sizes and the shared buffer space.

12 (Original). The apparatus of claim 11 wherein the network device's advertised window size calculation module further comprises:

a timer, initially set to a predetermined time interval Δt, and an iteration counter initially set to a predetermined initial value n;

a current queue size sampler that samples a current queue size $q_i(n)$ during the predetermined time interval Δt;

a current error signal calculation module that calculates a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

a window size calculation module that calculates the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

13 (Original). The apparatus of claim 12 wherein the current error signal calculation module further comprises:

a filter module that filters the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

wherein the window size calculation module calculates the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the

equation: $W_i(n) = \left[W_i(n-1) + \alpha\hat{e}_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

14 (Original). The apparatus of claim 11 wherein the feed back module further comprises:

an advertised window size updating module that updates a TCP receiver's advertised window size.

15 (Original). The apparatus of claim 14 wherein the advertised window size updating module further comprises:

an ACK packet identification module that identifies whether a packet is an ACK packet, and, if not, puts the non-ACK packet in a reverse data buffer;

an ACK packet classifier that determines a service class for the identified ACK packet;

an advertised window size reader that reads a TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

a window size comparison module that determines whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not sets an advertised window field in the identified ACK packet equal to the calculated network device's advertised window size $W_i(n)$ and updates the checksum field for the identified ACK packet.

16 (Original). The apparatus of claim 11 wherein the dynamic buffer threshold module further comprises:

a timer initially set to a predetermined time interval Δs and an iteration counter initially set to a predetermined initial value n;

a current queue size sampler that samples a current queue size $q_i(n)$ during the predetermined time interval Δs;

a current queue size calculation module that calculates a sum of the sampled current queue size according to the equation: $Q(n) = \sum_{i=1}^{K} q_i(n)$, where K is a number of service classes;

a dynamic buffer threshold determiner that determines whether the sum of the sampled current queue size is less than the product of a gain constant γ and a buffer size B;

and an updating module that updates the dynamic buffer threshold if the sum of the sampled current queue size is less than the product of the gain constant γ and the buffer size B, according to $\min\{T(n-1) + \Delta T, \gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes and if the sum of the sampled current queue size is not less than the product of a gain constant γ and a buffer size B, updates the dynamic buffer threshold according to $\max\{T(n-1) - \Delta T, T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold.

17 (Original). The apparatus of claim 16 wherein the current queue size calculation module further comprises:

a filter that filters the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n) = (1-\varphi)\hat{Q}(n-1) + \varphi Q(n)$, wherein φ is a predetermined parameter.

18 (Original). An article of manufacture for controlling the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, the article of manufacture comprising:

at least one processor readable carrier; and

instructions carried on the at least one carrier;

wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

organize a forward data buffer into one or more queues that store at least one forward data packet;

calculate a network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

provide the network device's advertised window size to a TCP source; and

calculate a dynamic buffer threshold based, at least in part, upon the sum of the queue sizes and the shared buffer space.

19 (Original). The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and

thereby cause the at least one processor to operate so as to:

organize the forward data buffer into one or more queues with one queue per service class.

20 (Original). The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

store the at least one forward data packet according to its service class.

21 (Original). The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

initialize a timer to a predetermined time interval Δt, and an iteration counter to a predetermined initial value n;

sample a current queue size $q_i(n)$ during the predetermined time interval Δt;

calculate a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

calculate the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters;

reset the timer, upon expiration of the predetermined interval Δt; and

iterate the iteration counter, upon expiration of the predetermined time interval Δt.

22 (Original). The article of manufacture of claim 21 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

filter the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

calculate the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha\hat{e}_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

23 (Original). The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

carry information relating to the network device's advertised window size by returning TCP acknowledgements in a receiver's advertised window field.

24 (Original). The article of manufacture of claim 18 wherein the instructions are configured to be readable from

the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

update a TCP receiver's advertised window size.

25 (Original). The article of manufacture of claim 24 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

identify whether a packet is an ACK packet, and, if not, put the non-ACK packet in a reverse data buffer;

determine a service class for the identified ACK packet;

read a TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

determine whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not setting an advertised window field in the identified ACK packet equal to the calculated network device's advertised window size $W_i(n)$ and updating the checksum field for the identified ACK packet.

26 (Original). The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and

thereby cause the at least one processor to operate so as to:

initialize a timer to a predetermined time interval Δs and an iteration counter to a predetermined initial value n;

set an initial dynamic buffer threshold $T(0)$ equal to a gain constant γ multiplied by a buffer size B and divided by a number of service classes K;

sample a current queue size $q_i(n)$ during the predetermined time interval Δs;

calculate a sum of the sampled current queue size according to the equation: $Q(n) = \sum_{i=1}^{K} q_i(n)$;

determine whether the sum of the sampled current queue size is less than the product of the gain constant and the buffer size γB;

if so, updating the dynamic buffer threshold according to $\min\{T(n-1) + \Delta T, \gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes;

if not, updating the dynamic buffer threshold according to $\max\{T(n-1) - \Delta T, T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold;

reset the timer, upon expiration of the predetermined interval Δs; and

iterate the iteration counter, upon expiration of the predetermined time interval Δs.

27 (Original). The article of manufacture of claim 26 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

filter the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n) = (1-\varphi)\hat{Q}(n-1) + \varphi Q(n)$, wherein φ is a predetermined parameter.

28 (Original). A signal embodied in a carrier wave and representing sequences of instructions which, when executed by at least one processor, cause the at least one processor to control the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, by performing the steps of:

organizing a forward data buffer into one or more queues that store at least one forward data packet;

calculating a network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

providing the network device's advertised window size to a TCP source; and

calculating a dynamic buffer threshold based, at least in part, upon the sum of queue sizes and the shared buffer space.

29 (Original). The signal of claim 28 wherein the step of organizing a forward data buffer further comprises:

organizing the forward data buffer into one or more queues with one queue per service class.

30 (Original). The signal of claim 28 wherein the at least one forward data packet is stored according to its service class.

31 (Original). The signal of claim 28 wherein the step of calculating a network device's advertised window size further comprises:

initializing a timer to a predetermined time interval Δt, and an iteration counter to a predetermined initial value n;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δt;

calculating a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters;

resetting the timer, upon expiration of the predetermined interval Δt; and

iterating the iteration counter, upon expiration of the predetermined time interval Δt.

32 (Original). The signal of claim 31 wherein the steps of

calculating a filtered current error signal $e_i(n)$ and calculating the network device's advertised window size further comprise:

filtering the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha\hat{e}_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

33 (Original). The signal of claim 28 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

carrying information relating to the network device's advertised window size by returning TCP acknowledgements in a receiver's advertised window field.

34 (Original). The signal of claim 28 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

updating a TCP receiver's advertised window size.

35 (Original). The signal of claim 34 wherein the step of updating a TCP receiver's advertised window size further comprises:

identifying whether a packet is an ACK packet, and, if not, putting the non-ACK packet in a reverse data buffer;

determining a service class for the identified ACK packet;

reading a TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

determining whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not setting an advertised window field in the identified ACK packet equal to the calculated network device's advertised window size $W_i(n)$ and updating the checksum field for the identified ACK packet.

36. (Original) The signal of claim 28 wherein the step of calculating a dynamic buffer threshold further comprises:

initializing a timer to a predetermined time interval Δs and an iteration counter to a predetermined initial value n;

setting an initial dynamic buffer threshold $T(0)$ equal to a gain constant γ multiplied by a buffer size B and divided by a number of service classes K;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δs;

calculating a sum of the sampled current queue size according to the equation: $Q(n) = \sum_{i=1}^{K} q_i(n)$;

determining whether the sum of the sampled current queue size is less than the product of the gain constant and the buffer size γB;

if so, updating the dynamic buffer threshold according to $\min\{T(n-1)+\Delta T, \gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes;

if not, updating the dynamic buffer threshold according to $\max\{T(n-1)-\Delta T, T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold;

resetting the timer, upon expiration of the predetermined interval Δs; and

iterating the iteration counter, upon expiration of the predetermined time interval Δs.

37 (Original). The signal of claim 36 wherein the step of calculating a sum of the sampled current queue size further comprises:

filtering the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n)=(1-\varphi)\hat{Q}(n-1)+\varphi Q(n)$, wherein φ is a predetermined parameter.

38 (Original). A computer data signal embodied in a carrier wave readable by a computing system and encoding a computer program of instructions for executing a computer process performing the method recited in claim 1.